

```

;
; ftt2.asm
; программное обеспечение реле-терморегулятора холодильного шкафа
;
; микроконтроллер AT89C2051, ЖКИ МТ10Т-7, датчик DS18B20,
; и силовой симисторный ключ с МОС3083
;
;
;
ORG 0000H
RST: AJMP INIT;вектор входа при аппаратном запуске или по INT0
      DS 1;в этом поле расположены или вектора или п/п по 3-5 байт!
;-----
;приведенные ниже метки должны быть заменены на соответствующие вектора
;при подгонке программ...
INT0: DS 8;вектор п/п прерывания от внешнего источника
TFI0: AJMP TXI0;вектор п/п прерывания от TC0 (таймер-счетчик TC0)
      DS 6
INT1: DS 8
TFI1: AJMP TXI1;вектор п/п прерывания от TC1 (таймер-счетчик TC1)
      DS 6
RTI: DS 16;вектор п/п прерывания от приемопередатчика
;-----
;блок объявленных имен :
;
; секция переопределения прямоадресуемых бит
;
ECRC: REG 2FH.0;флаг счетчика ошибок crc
ERD: REG 2FH.1;флаг ошибок "датчик не отвечает"/
      ;ошибка обмена канала связи с датчиком
EWFП: REG 2FH.2;ошибка записи EEPROM DS18B20
EOFF: REG 2FH.3;ошибка отключения (переохлаждение/перегрев)
ESNN: REG 2FH.4;ошибка клавиатуры
ERL: REG 2FH.5;ошибка CRC
CWK: REG 2EH.0;данные от кнопки готовы
CDT: REG 2EH.1;триггерный флаг обработчика данных DS18B20
      ;0 - в данном цикле выдать команду конвертации текущих параметров
      ;1 - в данном цикле считать и обработать предыдущие данные
BLWD: REG 2EH.2;блокировка обработчиков дисплея для маркеров
;программ обработки главных часов и термометра на время работы с
;клавиатурой
; ставится в 1 первой действительной кнопкой
; сбрасывается по выходу из основной "С" или основной "запись"
;в исполнительной части SKANY1 BLWD копируется в FSN1 (по умолчанию=0)
EROR: REG 2EH.3;флаг абсолютной ошибки (дополняется из ERVK)
SNNX: REG 2DH.0;подтверждение нажатия
SNNY: REG 2DH.1;подтверждение отпускания
SNNW: REG 2DH.2;команда к исполнению
FDOUT: REG 2CH.0;
FUS: REG 2CH.1;
FUS1: REG 2CH.2;
ZLL: REG 2CH.3;флаг активности тайм-аута повторного включения
; компрессора ( особенность термореле )
FSN2: REG 2CH.4;блокировка сканера клавиатуры на0.1-0.2S
; (совместно с SNER)
FSN3: REG 2CH.5;передаточный флаг включения во второй части
;поля указателей векторов кнопок
;устанавливается и/или сбрасывается в программах обработки текущих
;кодových комбинаций кнопок
WORK: REG 2CH.6;
MINUS: REG 2CH.7;флаг наличия символа "минус" в байте текущей
      ;температуры
FSN1: REG 2BH.4;флаг включения модифицированной части
;поля указателей векторов кнопок для п/п SKANY1

```

```

; в полной версии указатель берется из младшей тетрады регистра
; TABSN (формат 0000nnnn)
SN0: REG B0H.0;линии возврата клавиатуры S0-S3,могут быть
SN1: REG B0H.1; произвольно
SN2: REG B0H.2;переименованы в пределах выделенных ресурсов
SN3: REG B0H.3
SN4: REG B0H.4;действительно в jmp-версии задатчика
; уставки температуры
PMMD: REG B0H.7;префикс типа вводных данных (jmpreg(0) или snnx(1))
OUT: REG 90H.7;линия управления нагрузкой, активный уровень=0
DAT: REG 90H.6;линия обмена информацией с термодатчиком
ADR: REG 90H.5;линия A0 блока LCD
WLD: REG 90H.4;линия W\ блока LCD

```

```

;
;
;
;
-----
секция переопределения регистров РСФ/РПД

```

```

ADLCD: REG 18H;адрес стартовой позиции ОЗУ LCD
CTLCD: REG 1AH;счетчик количества байт в блоке обмена с LCD
TABSN: REG 2AH;указатель статуса рабочего поля клавиатуры
; для режимов "многослойных" вложений в полной версии программы
;
SNNK: REG 2BH;копия текущего статуса P3
FLAG: REG 2CH;регистр флаг- бит
SNVK: REG 2DH;смещение указателя вектора клавиатуры
WRVK: REG 2EH;смещение указателя вектора текущей процедуры
; диспетчера задач
ERVK: REG 2FH;смещение указателя вектора текущей процедуры
; обработчика ошибок
;
;
; информационный файл DS18B20
TERML: REG 30H;данные температуры L-байт
TERMH: REG 31H;данные температуры H-байт
HTPU: REG 32H;уставка верхнего порога тревоги
LTPU: REG 33H;уставка нижнего порога тревоги
CONFIG: REG 34H;конфигурация датчика,прочитанная при очередном цикле
;чтения данных из датчика
RV0: REG 35H;резерв - должно быть #ffh
RV1: REG 36H;резерв - должно быть #ffh
RV2: REG 37H;резерв - должно быть #ffh
CNK: REG 38H;байт циклического кода
;
;
ATBSNL: REG 39H;регистры хранения текущей таблицы векторов обработчика
ATBSNH: REG 3AH;клавиатуры
CNTSL: REG 3BH;счетчик досчета тайм-паузы L-байт (залипание)
CNTSH: REG 3CH;счетчик досчета тайм-паузы H-байт (залипание)
STTBL: REG 3DH; предварительные значения для адреса начала
STTBH: REG 3EH; текущей таблицы обработчиков кнопок для DPTR
TEMP: REG 3FH;регистр временного хранения данных для процедур
;настройки уставок
DISP0: REG 40H;поле ОЗУ дисплея 0-9 и F
DISP1: REG 41H
DISP2: REG 42H
DISP3: REG 43H
DISP4: REG 44H
DISP5: REG 45H
DISP6: REG 46H
DISP7: REG 47H
DISP8: REG 48H
DISP9: REG 49H
DISPF: REG 4FH;байт разблокировки дисплея
TMPSN: REG 50H; PBX текущего защелкнутого кода
TMPSVL: REG 51H;
TMPSVH: REG 52H;

```

CNER: REG 53H;счетчик непрерывных ошибок канала обмена с датчиком
 CONF: REG 54H;источник/дублер значения текущей конфигурации
 ;датчика (для ds18b20)
 ;загружается требуемым значением при выполнении init-блока
 TMP1: REG 55H;байт=текущая Т , целое число со знаком (-128 +127)
 LTPUD: REG 56H;
 HTPUD: REG 57H;
 FTEST: REG 58H;регистр заданной температуры (знаковая величина)
 ; используется в основном для тест-отладки
 CZLL: REG 59H;пауза досчета перезапуска с учетом термореле
 SNER: REG 5AH; второй счетчик (внешний) тайм-аута "залипания"
 SNER1: REG 5BH; внутренний счетчик тайм-аута "залипания"
 CMSN: REG 5CH;счетчик паузы автовозврата

```

;-----
;
; сетка опроса клавиатуры (при активации CWK)
TXI0: CLR TR0; =0,05s
      CLR T1
      MOV TL0,#B0H
      MOV TH0,#3CH
      SETB CWK; временно блокируется на тестах терморегулятора
      SETB T1
      SETB TR0
      RETI
;-----

```

```

;-----
;
; сетка опроса датчика\цикл регулирования (при активации CDT)
TXI1: CLR TR1; =2s (=5s)
      MOV TL1,#D8H;9CH
      MOV TH1,#FFH
      SETB CDT
      SETB TR1
      RETI
;-----

```

```

;-----
INIT: MOV R0,#7FH
      CLR A
LOOP0: MOV @R0,A
      DJNZ R0,LOOP0;очистка озу сри
      MOV SP,#5FH;переустановка указателя стека
      MOV DPTR,#VEKTS
      MOV TMOD,#51H
      MOV IE,#0AH
      MOV TL0,#B0H; =0,05S
      MOV TH0,#3CH
      MOV TL1,#D8H; =2S (9CH; =5S)
      MOV TH1,#FFH
      MOV CONF,#1FH;конфигурировать датчик на 9-битовый режим
      ;измерения температуры (для работы с ds18b20)
      MOV CNER,#3;начальная загрузка счетчика лимита ошибок CRC
      ;MOV FTEST,#-5D;10D;#F6H; -10D
      MOV DISPF,#FFH;байт разблокировки дисплея
      ACALL RESLCD;инициализация дисплея
      ACALL TYTT
;-----

```

```

;-----
;
; сервисный контроль и загрузка начальных параметров
; термодатчика
;-----

```

```

TEST1: ACALL DTTC;генерация импульса сброса/предустановки датчика
        ACALL T110;защитный интервал ~10мс
        MOV A,#CCH;"ИГНОРИРОВАТЬ ПЗУ"
        ACALL TXB;защитный интервал ~10мс вмонтирован в TXB
        MOV A,#44H;стартовая команда "конверсия температуры"
        ACALL TXB
        ACALL DTTC
        ACALL T110

```

```

JNB ERD,GOTS1
SETB ERROR;приводит диспетчер задач в режим вывода состояния
;отображения ошибки
AJMP EROLL;обработчик ошибок
;вывод строки "нет ответа датчика" и активация соотв. флагов
GOTS1: SETB EA;разрешить прерывания
SETB TR1;пуск C\T1
SETB TR0;пуск C\T0
; сетка таймера диспетчера задачи активирована
;-----
;ЗАГРУЗКА НАЧАЛЬНОГО СООБЩЕНИЯ В ДИСПЛЕЙ
MOV R4,#<STR2
MOV R5,#>STR2
ACALL SINDI
ACALL LLCD10
ACALL TYTT
;-----
;
TRPP: SETB ZLL;защитная пауза термореле при перезапуске
MOV CZLL,#50D;питания системы
;-----
;
; модуль обработки диспетчера задачи /WAIT_FLAG/
;
;
PATCH: NOP
NOP
MOV A,WRVK
ADD A,#1
MOVC A,@A+PC
JMP @A+DPTR
VPATCH: DB RELY-VEKTS;00000000 режим ожидания очередного кванта ITFn
DB SKAS0-VEKTS;00000001 CWK=1 выполнить обработчик клавиатуры
DB WCDT-VEKTS;00000010 CDT=1 выполнить обработчик данных
; терморегулятор и/или часы
DB WCDT-VEKTS;00000011 ; выполнить обработчик данных и их
;вывод на дисплей (ОЗУ дисплея + LLCD10 + CWK=0), если BLWD=0
;при BLWD=1 вывод данных на дисплей (ОЗУ дисплея + LLCD10)
; НЕ ВЫПОЛНЯЕТСЯ!
DB RELY0-VEKTS;00000100 режим ожидания очередного кванта ITFn
;в активной панели настройки
DB SKAS0-VEKTS;00000101;клавиатура в активной панели настройки
DB RELY0-VEKTS;00000110;неисполнимый CDT
;в активной панели настройки (со сбросом CDT=0)
DB RELY0-VEKTS;00000111;клавиатура в активной панели настройки
;аналог неисполнимого CDT в активной панели
;настройки (со сбросом CDT=0)
DB WKER-VEKTS;00001000 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001001 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001010 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001011 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001100 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001101 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001110 ERROR=1 обработчик ошибок
DB WKER-VEKTS;00001111 ERROR=1 обработчик ошибок
;-----
;
RESLCD: MOV ADLCD,#4FH;активация дисплея при включении питания
MOV CTLCD,#1
SJMP LLCD
;
;-----
LLCD10: MOV ADLCD,#40H;загрузка строки кодов символов из ОЗУ в LCD
MOV CTLCD,#10D;сразу все 10 позиций
;основной модуль перезагрузки LCD
;перед вызовом загрузить начальный адрес блока данных в ADLCD
;количество пересылаемых в данной сессии байт в CTLCD
;используется регистровый банк rb3

```

;код сегментов, пересылаемых в LCD должен быть предварительно размещен
 ;в соответствующих ячейках ОЗУ (40h=крайняя левая позиция, 49h=
 ;крайняя правая позиция) код активации дисплея = ffh(в ячейке 4fh)

;

```
LLCD:  PUSH A
        PUSH PSW
        SETB RS0
        SETB RS1
        MOV R5,#F0H
        MOV R4,#0FH
        MOV R3,#FFH
```

```
OUTADR: MOV A,R4
        ANL A,R0
        ORL A,R5
        MOV P1,A
        CLR ADR
        CLR WLD
        SETB WLD
        SETB ADR
        MOV P1,R3
```

```
OUTDAT: MOV A,@R0
        ACALL TETR
        MOV A,@R0
        SWAP A
        ACALL TETR
        INC R0
        DJNZ R2,OUTDAT
        POP PSW
        POP A
        RET
```

```
TETR:  ORL A,R5
        MOV P1,A
        CLR WLD
        SETB WLD
        MOV P1,R3
        RET
```

;

; предварительная загрузка строки информационного сообщения
 ; в ОЗУ дисплея с последующим выводом на экран
 ; всего информационных байт - 250
 ; старший байт DPTR загружается в SINDI, а младший должен быть передан
 ; в аккумуляторе перед вызовом SINDI (

```
;mov r4,#<strnn
;mov r5,#>strnn
;acall sndi      )
```

;

```
SINDI: PUSH DPL
        PUSH DPH
        MOV DPH,R5;
        MOV DPL,R4;
        MOV R0,#DISP0
        MOV R2,#10D
```

```
LPIND:  CLR A
        MOVC A,@A+DPTR
        MOV @R0,A
        INC DPTR
        INC R0
        DJNZ R2,LPIND
        POP DPH
        POP DPL
        RET
```

;

; подпрограмма знакогенератора стандартного ряда 0 - F символов для дисплея

```

; активный уровень = 1
; код для преобразования в формате 0000xxxxB должен быть помещен в аккумулятор
; при выходе в A код символа
; используется текущий PБ
    ZNAK: INC A
           MOV C,A,@A+PC
           RET
           DB EEh,60h,2Fh,6Dh,E1h,CDh,CFh,68h
           DB EFh,EDh,EBh,C7h,8Eh,67h,8Fh,8Bh
;*****
; служебные подпрограммы блока обмена информацией с термодатчиком
; DTTC - генератор импульса reset термодатчика и проверка импульса
; "присутствия"
    DTTC: CLR DAT; сбросить линию связи с термодатчиком в 0
           MOV R3,#2
    T500: MOV R2,#F0H
    T501: DJNZ R2,T501
           DJNZ R3,T500
           SETB DAT; отработка импульса сброса(~500mkS)
           MOV R3,#17D; загрузить счетчик активного интервала опроса
    T00:  MOV R2,#10H; тайм-аут 10мс
    T10:  DJNZ R2,T10; проверка отклика через каждые 9 мкс за 170 мкс
           JNB DAT,T02; ловушка "нуля" отклика датчика
           DJNZ R3,T00
           SETB ERD; ошибка "датчик не отвечает"
           RET
    T02:  JNB DAT,T02; ждать конец импульса присутствия
           RET
;-----
; RXB - чтение байта информации из датчика
; вызов acall gxb
; по выходу принятый байт в регистре A
    RXB:  MOV R4,#8
    LOOPR1: CLR DAT
            NOP
            NOP
            SETB DAT; ~1,5 mkS
            MOV R2,#2
    L000:  DJNZ R2,L000
            MOV C,DAT
            RRC A; текущий бит в A
            ACALL T60
            ACALL T110
            DJNZ R4,LOOPR1
            RET
;-----
; TXB - передача байта информации в датчик
; перед вызовом передаваемый байт разместить в A
; вызов acall txb
    TXB:  MOV R4,#8
    LOOPT1: CLR DAT
            NOP
            NOP
            NOP
            RRC A
            MOV DAT,C
            ACALL T90
            SETB DAT
            ACALL T110
            DJNZ R4,LOOP1
            RET
;-----
; CRC - КОНТРОЛЬ

```

```

CRCK: CLR ERL
      CLR A
      MOV R7,A
      MOV R0,#30H
      MOV R2,#9
CRCL: MOV A,@R0
      MOV R5,A;PUSH A
      MOV R3,#8
CRCLP: XRL A,R7;XRL A,TEMP
       RRC A
       MOV A,R7;MOV A,TEMP
       JNC ZERO
       XRL A,#18H
ZERO:  RRC A
       MOV R7,A;MOV TEMP,A
       MOV A,R5;POP A
       RR A
       MOV R5,A;PUSH A
       DJNZ R3,CRCLP
       INC R0
       DJNZ R2,CRCL
       CLR A
       ADD A,R7
       JZ WCRC
       SETB ERL
WCRC: RET

```

```

;-----
;данный фрагмент предназначен для нормирования данных до целого числа
;при 9-битовом формате с датчиком DS18B20

```

```

NORM9: MOV R0,#TERMH
       MOV A,TERML
       XCHD A,@R0
       SWAP A
       MOV TMP1,A;данные с учетом знака в TMP1
       RET

```

```

;-----
;*****
;
; блок генерации временных интервалов п/п обмена с термодатчиком
;-----

```

```

; генератор ~10мкс
T110: MOV R2,#5D
T110L: DJNZ R2,T110L
      RET

```

```

;-----
; добавка для временного слота чтения
T60:  MOV R2,#55D
T60L: DJNZ R2,T60L
      RET

```

```

;-----
; добавка для временного слота записи
T90:  MOV R2,#53D
T90L: DJNZ R2,T90L
      RET

```

```

;*****
;
;-----

```

```

;
; VEKTS общая таблица команд переходов на п/п обслуживания
; диспетчера задач, сканера клавиатуры, кнопок клавиатуры
; (директив управления)
;

```

```

;
; блок векторов сканера клавиатуры
VEKTS: DS 2
SKAS0: AJMP SKAS
LOXS:  AJMP PATCH;RET

```

```

BLEFS: AJMP BLEFS1
;
;-----
;
; блок векторов директив обнаружения нажатия
; обработчики кнопок лист 0
;
S0_0: AJMP S0_00;активация уставки 0C
S1_0: AJMP S1_00;активация настройки уставок в диапазоне +1 - +30C
S2_0: AJMP S2_00;активация настройки уставок в диапазоне -1 - -30C
S3_0: AJMP S3_00;просмотр текущей уставки
;
;-----
;
; блок векторов директив обнаружения нажатия
; обработчики кнопок лист 1
;
S0_1: AJMP S0_10;старший разряд +1
S1_1: AJMP S1_10;младший разряд +1
S2_1: AJMP S2_10;ввод параметров
S3_1: AJMP S3_10;сброс режима клавиатуры
;
;-----
;
; блок векторов директив диспетчера задач
;
WCDT: AJMP TERMS;чтение текущих данных и их обработка
; дать команду "конвертировать Т"
; обработка OUT, вывод на дисплей, если BLWD=0
WKER: AJMP EROLL;обработчик ошибок
SKANS0: AJMP SKANS1
SKANX0: AJMP SKANX1
SKANY0: AJMP SKANY1
RELY: AJMP PATCH;сканирование на "холостом ходу"
RELY0: AJMP RELY01
;-----
;
; основной обработчик терморегулятора TERMS
;
TERMS: CLR CDT;отмена повторного входа в процедуру из PATCH
CLR CWK;обработчик клавиатуры игнорируется при BLWD=0
MOV ERVK,#0;очистка регистра ошибок
ACALL RDAT;обмен с датчиком
CLR A
ADD A,ERVK;регистр ошибок=0?
JZ WORGS;при отсутствии ошибок продолжить от WORGS
JNB ECRC,ETRS;при ошибках не группы CRC продолжить от ETRS
DJNZ CNER,ENDTRS;контроль доверия CRC (следующая попытка)
ETRS: AJMP EROLL
ENDTRS: AJMP PATCH
;-----
;
WORGS: MOV CNER,#3;перезагрузка счетчика доверия
ACALL NORM9;текущая Т со знаком нормированная до байта в TMP1
JNB ZLL,GORMS;обработчик паузы термореле
DJNZ CZLL,SRMS
CLR ZLL
GORMS: ;MOV A,FTEST;уставка отключения считана из FTEST
MOV A,HTPU;уставка отключения считана из DS18B20 в HTPU
;-----
;
CJNE A,#0,RPP1;анализ на уставку=0
AJMP ZERU;перейти к обработчику уставки=0
;-----
;
RPP1: JNB A.7,RPP0;обработка положительной уставки
AJMP RMINS;обработка отрицательной уставки
RPP0: AJMP RPLUS
;-----
;
SRMS: MOV A,TMP1
JNB A.7,SRMS1;обработка положительной температуры
; в паузе термореле

```



```

        ACALL HABS;обработка отрицательной температуры
        ; в паузе термореле
        ACALL OZUWR
        MOV DISP4,#1; "-"
SRMS3: MOV DISP2,#10H;признак задержки термореле
        ;(возможен другой символ)
SRMS2: MOV DISP1,#47H;нижний кружок
        ACALL LLCD10
        SETB OUT
        AJMP PATCH
SRMS1: ACALL OZUWR
        MOV DISP4,#0
        SJMP SRMS3
;-----
RMINS: MOV A,TMP1
        JB A.7,RMS0
;-----
;заведомо "+" температура
        ACALL OZUWR
        MOV DISP4,#0
RMS2:  MOV DISP1,#A9H;верхний кружок
        ACALL LLCD10
        CLR OUT
        AJMP PATCH
;-----
RMS0:  ;MOV A,FTEST
        MOV A,HTPU
        ACALL HABS
        MOV R7,A
        MOV A,TMP1
        ACALL HABS
;-----
RMS1:  CJNE A,07H,RMS3; T<>Tu
        SJMP RMS5; T=Tu (-Tu) OFF
RMS3:  JC RMS6; T<Tu (-Tu) ON
;-----
RMS5:  SETB ZLL; T>Tu (-Tu) OFF
        MOV CZLL,#50D
        ACALL OZUWR
        MOV DISP4,#1
        MOV DISP1,#47H;нижний кружок
        ACALL LLCD10
        SETB OUT
        AJMP PATCH
;-----
RMS6:  ACALL OZUWR;вывод данных при T<Tu (-Tu)
        MOV DISP4,#1
        SJMP RMS2
;-----
RPLUS: MOV A,TMP1
        JNB A.7,RPLS0
        ACALL HABS
        SJMP RMS5;    T<0!
RPLS0: CJNE A,#0,RPLS1
RPLS3: SETB ZLL;    T=0!
        MOV CZLL,#50D
        ACALL OZUWR
        MOV DISP4,#0
        AJMP SRMS2
RPLS1: ;CJNE A,FTEST,RPLS2
        CJNE A,HTPU,RPLS2
        SJMP RPLS3;    T=Tu
RPLS2: JC RPLS3;    T<Tu

```

```

ZU1:  ACALL OZUWR;      T>Tu
      MOV DISP4,#0
      MOV DISP1,#A9H;верхний кружок
      ACALL LLCD10
      CLR OUT
      AJMP PATCH
;-----
;
ZERU:  MOV A,TMP1
      JNB A.7,ZU0;величины от 0 до -1
      ;автоматически приравиваются к -1 (ACALL NORM9)
      ACALL HABS
      AJMP RMS5
ZU0:  SJMP ZU1
;-----
;
HABS:  CPL A;получение абсолютной величины
      ADD A,#1; отрицательного числа
      RET
;-----
;
; декодировка и ввод данных в ОЗУ дисплея
; перед вызовом загрузить в А байт абсолютной величины данных!
OZUWR: ACALL XCL
      MOV A,R5
      ACALL ZNAK
      MOV DISP6,A;сегментный код единиц в DISP7
      MOV A,R6
      CJNE A,#0,OZWR0
      MOV DISP5,#0
      SJMP OZWR1
OZWR0: ACALL ZNAK
      MOV DISP5,A;сегментный код десятков в DISP6
OZWR1:  MOV DISP8,#8EH
      MOV DISP7,#A9H
      CLR A;очистка общего поля
      MOV DISP9,A
      MOV DISP4,A
      MOV DISP3,A
      MOV DISP2,A
      MOV DISP1,A
      MOV DISP0,A
      RET
;-----
;
; блок строк информационных сообщений
; позиции отображения -0-1-2-3-4-5-6-7-8-9-
;
TBINF: DB FFH,FFH,FFH,FFH,FFH,FFH,FFH,FFH,FFH,FFH;тест "все включено"
STR1:  DB 0H,0H,0H,0H,0H,0H,0H,0H,0H,0H;тест "все отключено"
STR2:  DB EEh,60h,2Fh,6Dh,E1h,CDh,CFh,68h,EFh,EDh;тест "0123456789"
STR3:  DS 10;DB
STR4:  DB 1H,0H,1H,0H,1H,0H,1H,0H,1H,0H; "- - - - -"
STR5:  DB 00H,8FH,03H,EEH,03H,00H,04H,00H,04H,00H; " ErOr _ _"
STR6:  DB 00H,EBH,67H,66H,00H,00H,00H,00H,A9H,8EH; " Adj  'C"
STR7:  DB 00H,00H,00H,00H,00H,00H,A9H,8EH; "    'C "
STR8:  DB 47H,04H,04H,8DH,4EH,2DH,04H,04H,04H,47H;стилизация "запись"
STR9:  DB 00H,8BH,E6H,CDH,00H,00H,00H,00H,A9H,8EH; " FUS  'C"
;-----
;преобразователь байта положительного числа в двоично-десятичном формате из
;аккумулятора в три байта в упакованном десятичном формате, размещаемых в
;регистрах r5, r6 и r7 текущего банка регистров
XCL:  XCH A,R7;хранить текущий А в r7
      CLR A;очистка регистров данных
      MOV R5,A;
      MOV R6,A;
      XCH A,R7;
      CJNE A,#0,GOXL;выполнять только при А не равном 0

```

```

        RET;(по условиям обработки в А только положительное число)
GOXL:  CJNE A,#100D,KONT1;если А не равно 100, то KONT1
        SJMP PGM1;если А=100, то PGM1
KONT1:  JC KONT2;если А<100,то KONT2
PGM1:   MOV B,#100D;делитель в В
        DIV AB;
        MOV R7,A;частное в R7 ( сотни в R7 )
        MOV A,B;остаток в А
KONT2:  CJNE A,#10D,KONT3;если А не равно 10, то KONT3
        SJMP PGM2;если А=10, то PGM2
KONT3:  JC EPGM;если А<10,то EPGM
PGM2:   MOV B,#10D;делитель в В
        DIV AB;
        MOV R6,A;частное в R6 ( десятки в R6 )
        MOV A,B;остаток в А
EPGM:   MOV R5,A;единицы в R5
        RET
;-----
;
;      обработчик текущей ошибки
;
;
EROLL:  SETB OUT;нагрузка отключена
        MOV R4,#<STR5
        MOV R5,#>STR5
        ACALL SINDI
        MOV DISP7,ERVK
        ACALL LLCD10
        SETB ERROR
;-----
;
ERSTP:  SJMP ERSTP
;-----
;
        AJMP PATCH
;-----
;
;модуль обмена информацией с термодатчиком
;
;
RDAT:   ACALL RDATU
        CLR TR0
        CLR TR1
        CLR EA;запрет всех прерываний на время работы с термодатчиком
        ACALL DTTC;генерация импульса сброса/предустановки датчика
        ACALL T110;защитный интервал ~10мкс
        MOV A,#ССН;"ИГНОРИРОВАТЬ ПЗУ"
        ACALL TXB;защитный интервал ~10мкс вмонтирован в TXB
        MOV A,#44H;"ПРЕОБРАЗОВАНИЕ ТЕМПЕРАТУРЫ"
        ACALL TXB;защитный интервал ~10мкс вмонтирован в TXB
        SETB EA
        SETB TR1
        SETB TR0
        RET
;-----
;
RDATU:  ;ACALL KONTDТ;контроль линии "DAT" на короткое замыкание
        CLR TR0
        CLR TR1
        CLR EA;запрет всех прерываний на время работы с термодатчиком
        ACALL DTTC
        ACALL T110;защитный интервал ~10мкс
        MOV A,#ССН;"ИГНОРИРОВАТЬ ПЗУ"
        ACALL TXB;защитный интервал ~10мкс вмонтирован в TXB
        MOV A,#ВЕН;"ЧИТАТЬ ОЗУ ДАТЧИКА"
        ACALL TXB;защитный интервал ~10мкс вмонтирован в TXB
        ;читать блок данных из термодатчика в буферное озу,
READT:  MOV R0,#30H;размещенное в рпд контроллера в R_30 - R_38
        MOV R5,#09H;всего 9 байт информации
RXDT:   ACALL RXB

```

```

MOV @R0,A
INC R0
DJNZ R5,RXDT
ACALL DTTC
ACALL CRCK
SETB EA
SETB TR1
SETB TR0
RET

```

```

;-----
; селектор обработчиков кнопок
; вызывается обработчиком PATCH при векторе с CWK=1
; проводится дополнительная обработка векторного перехода
; по содержимому SNVK
; (возможна обработка таблицы с учетом 4-х флагов: SNNX,SNNY,SNNW и INIS)

```

```

SKAS: MOV A,SNVK
      ADD A,#1
      MOVC A,@A+PC
      JMP @A+DPTR
TXSKAS: DB SKANS0-VEKTS; 0000
        DB SKANX0-VEKTS; 0001
        DB SKANY0-VEKTS; 0010
        DB RELY-VEKTS; 0011 БЛОКИРОВАННЫЕ КОМБИНАЦИИ
        DB RELY-VEKTS; 0100
        DB RELY-VEKTS; 0101
        DB RELY-VEKTS; 0110
        DB RELY-VEKTS; 0111
        DB RELY-VEKTS; 1000
        DB RELY-VEKTS; 1001
        DB RELY-VEKTS; 1010
        DB RELY-VEKTS; 1011
        DB RELY-VEKTS; 1100
        DB RELY-VEKTS; 1101
        DB RELY-VEKTS; 1110
        DB RELY-VEKTS; 1111

```

```

;-----
; стандартная процедура ожидания нажатия кнопок
;исходно SNNK=0, SNVK=0 PATCH обрабатывает вектор с CWK=1
;

```

```

SKANS1: CLR CWK;запрет повторного выполнения
        JNB FSN1,GOSNX3;при FSN1=1 отработать защитный интервал 0.3S
        DJNZ SNER,GOSNX2;
        CLR FSN1; FSN1=0 SNER=0
GOSNX3: ACALL LVNP
        ADD A,#0;проверка на наличие активности
        JNZ GOSNX;при обнаружении активности продолжить от GOSNX
GOSNX2: CLR A;иначе
        MOV SNNK,A;
        MOV SNVK,A;
GOSNX1: AJMP PATCH;SNNK=SNVK=0 и возврат в точку PATCH
GOSNX:  MOV SNNK,A;хранить текущее значение ЛБК в SNNK
        SETB SNNX; активировать флаг SNNX
        SJMP GOSNX1;продолжить от GOSNX1

```

```

;-----
; стандартная процедура ожидания подтверждения нажатия кнопок
;исходно SNNK=код обнаруженной комбинации
; SNNX=1 (SNVK=01H) PATCH обрабатывает вектор с CWK=1
;

```

```

SKANX1: CLR CWK;запрет повторного выполнения
        ACALL LVNP
        CJNE A,SNNK,ESNX;переход по ошибке антидребезга
        MOV SNER1,#C8H;константы интервала "ошибки залипания"
        MOV SNER,#02H; 20S при кванте=0.05S

```

```

CLR SNNX
SETB SNNY;активировать флаг ожидания отпускания
AJMP PATCH
ESNX: SJMP GOSNX2;очистка SNNK=SNVK=0 с выходом в PATCH
;-----
; стандартная процедура ожидания отпускания кнопок
;исходно SNNK=код обнаруженной комбинации
; SNNX=0 SNNY=1 (SNVK=02H) PATCH обрабатывает вектор с CWK=1
;
SKANY1: CLR CWK;запрет повторного выполнения
ACALL LVNP
ADD A,#0;проверка на наличие активности
JZ DSNX;при обнаружении отпускания продолжить от DSNX
DJNZ SNER1,WAITS2;отработка паузы "залипания"
DJNZ SNER,WAITS1
SETB ESNN;флаг "ошибка клавиатуры"
SETB EROR;флаг активации обработчика ошибок
WAITS2: AJMP PATCH
WAITS1: MOV SNER1,#C8H;перезагрузка внутреннего счетчика "залипания"
SJMP WAITS2
DSNX: MOV SNER,#5D;модуль защиты от повторного сканирования 0,25S
SETB FSN2
MOV SNVK,#0;очистка указателя
MOV C,BLWD;читать текущий префикс статуса в C
MOV FSN1,C;установить префикс статуса в текущем коде SNNK
MOV A,SNNK;код текущей комбинации в A
ADD A,#1
MOVC A,@A+PC;считать указатель из таблицы VSNX
JMP @A+DPTR;отработать модуль по VEKTS+указатель
;-----
;где Snn -имя модуля обработки кодовой комбинации
;ПОЛЕ ПО УМОЛЧАНИЮ
VSNX: DB LOXS-VEKTS;Snn-VEKTS;00000000 ничего нет
DB S0_0-VEKTS;00000001 S0
DB S1_0-VEKTS;00000010 S1
DB LOXS-VEKTS;Snn-VEKTS;00000011
DB S2_0-VEKTS;00000100 S2
DB LOXS-VEKTS;Snn-VEKTS;00000101
DB LOXS-VEKTS;Snn-VEKTS;00000110
DB LOXS-VEKTS;Snn-VEKTS;00000111
DB S3_0-VEKTS;00001000 S3
DB LOXS-VEKTS;Snn-VEKTS;00001001
DB LOXS-VEKTS;Snn-VEKTS;00001010
DB LOXS-VEKTS;Snn-VEKTS;00001011
DB LOXS-VEKTS;Snn-VEKTS;00001100
DB LOXS-VEKTS;Snn-VEKTS;00001101
DB LOXS-VEKTS;Snn-VEKTS;00001110
DB LOXS-VEKTS;Snn-VEKTS;00001111
;ПОЛЕ С BLWD=FSN1=1
VSNX1:DB LOXS-VEKTS;Snn-VEKTS;00010000 ничего нет
DB S0_1-VEKTS;00010001 S0
DB S1_1-VEKTS;00010010 S1
DB LOXS-VEKTS;Snn-VEKTS;00010011
DB S2_1-VEKTS;00010100 S2
DB LOXS-VEKTS;Snn-VEKTS;00010101
DB LOXS-VEKTS;Snn-VEKTS;00010110
DB LOXS-VEKTS;Snn-VEKTS;00010111
DB S3_1-VEKTS;00011000 S3
DB LOXS-VEKTS;Snn-VEKTS;00011001
DB LOXS-VEKTS;Snn-VEKTS;00011010
DB LOXS-VEKTS;Snn-VEKTS;00011011
DB LOXS-VEKTS;Snn-VEKTS;00011100
DB LOXS-VEKTS;Snn-VEKTS;00011101

```

DB LOXS-VEKTS;Snn-VEKTS;00011110
DB LOXS-VEKTS;Snn-VEKTS;00011111

```
;-----
TYTT: MOV R3,#FFH;задержка на внутренние процессы дисплея
TY1:  MOV R2,#FFH
TY0:  DJNZ R2,TY0
      DJNZ R3,TY1
      RET
;-----
LVNP:  MOV A,#F0H;маска в A
      ORL A,P3;ввод текущего состояния P3 в A с выделением пакета ЛБК
      CPL A;инверсия для получения активного уровня ЛБК=1
      RET
;-----
; возврат в процедурах PATCH при CWK=0 и BLWD=1
; тайм-аут автовозврата из режима клавиатуры
BLEFS1: DJNZ CMSN,DNSX
        AJMP S3_1;кнопка C режим 1
DNSX:   AJMP PATCH
;-----
;общая часть всех блоков активации клавиатуры
;
START0: ACALL STTR0
        MOV R4,#<STR6;строка заставки режима в ОЗУ дисплея
        MOV R5,#>STR6
        ACALL SINDI
        RET
STTR0:  SETB BLWD;блокировать обработчик терморегулятора
        SETB OUT;отключить нагрузку
        SETB ZLL;активировать режим паузы термореле
        MOV CZLL,#50D;
        MOV CMSN,#150D;загрузить счетчик паузы автовозврата
        RET
;-----
;КЛАВИАТУРА стандартно:
;
;      S0 активация ввода уставки 0C
;      S1 активация настройки уставок от +1C до +30C
;      S2 активация настройки уставок от -1C до -30C
;      S3 просмотр текущей уставки
;
; второе поле:
;      S0 текущий старший разряд +1
;      S1 текущий младший разряд +1
;      S2 ввод параметров
;      S3 возврат в режим регулирования без внесения изменений
;-----
; S0 стандартно активация ввода уставки 0C
S0_00: ACALL START0
        MOV DISP6,#EEH;запись символа 0 в DISP6
        CLR A
        MOV TMP5VH,A;rvx старшей тетрады задатчика уставки
        MOV TMP5VL,A;rvx младшей тетрады задатчика уставки
        SETB FSN3;кнопки S0 и S1 в этом режиме не обрабатываются
        ACALL LLCD10;вывод текущей информации в дисплей
        AJMP PATCH;возврат в диспетчер задач
;-----
; S1 стандартно активация настройки уставок от +1C до +30C
S1_00: ACALL START0
        CLR MINUS
GOU1:  ACALL REKS
        MOV A,HTPU;уставка в данной версии знаковый байт из HTPU
        JNB A.7,GOU0;при положительном HTPU от GOU0
        ACALL HABS;иначе получить абсолютную величину
;-----
```

```

;декодировка и вывод текущего значения на индикацию
;
GOU0: ACALL XCL;
      MOV A,R5;
      MOV TMPSVL,A
      ACALL ZNAK
      MOV DISP6,A
      MOV A,R6
      MOV TMPSVH,A
      ACALL ZNAK
      MOV DISP5,A
      ACALL LLCD10
      AJMP PATCH
;-----
; S2 стандартно активация настройки уставок от -1C до -30C
S2_00: ACALL START0
      SETB MINUS
      MOV DISP4,#1;символ "-"
      SJMP GOU1
;-----
; S3 стандартно просмотр текущей уставки
S3_00: ACALL STTR0
      MOV R4,#<STR9;строка заставки режима в ОЗУ дисплея
      MOV R5,#>STR9
      ACALL SINDI
      SETB FUS;запрет кнопки "ввод"
      SETB FSN3;кнопки S0 и S1 в этом режиме не обрабатываются
      ACALL REKS
      MOV A,HTPU;уставка в данной версии знаковый байт из HTPU
      JNB A.7,GOU0;при положительном HTPU от GOU0
      ACALL HABS;иначе получить абсолютную величину
      SETB MINUS
      MOV DISP4,#1;символ "-"
      SJMP GOU0
;-----
; S0 второе поле текущий старший разряд +1
S0_10: MOV R0,#TMPSVH
      MOV R1,#DISP5
GOU5: MOV CMSN,#150D;перезагрузка счетчика автовозврата
      JNB FSN3,GOU4
      AJMP PATCH
;-----
GOU4: MOV A,@R0
      INC A
      DA A
      ANL A,#0FH
      MOV @R0,A
      ACALL ZNAK
      MOV @R1,A
      ACALL LLCD10
      AJMP PATCH
;-----
; S1 второе поле текущий младший разряд +1
S1_10: MOV R0,#TMPSVL
      MOV R1,#DISP6
      SJMP GOU5
;-----
; S2 второе поле ввод параметров
S2_10: JNB FUS,S2GO
      AJMP PATCH
S2GO: MOV A,TMPSVH
      ANL A,#3;маска допуска диапазона от +30 до -30
      MOV B,#10D

```

```

MUL AB
ADD A,TMP SVL
MOV R5,A
MOV HTPUD,A
CLR C
SUBB A,#2
MOV LTPUD,A
JNB MINUS,SNVR1
MOV A,HTPUD
CPL A
ADD A,#1
MOV HTPUD,A
ADD A,#2
MOV LTPUD,A
SNVR1: ACALL SNWRI;далее обработка S3_10
MOV R4,#<STR8
MOV R5,#>STR8
ACALL SINDI
MOV R6,#FFH
MTM: ACALL TYTT
DJNZ R6,MTM
;-----
; S3 второе поле возврат в режим регулирования без внесения изменений
S3_10: CLR FUS;сброс префикса защиты из S3_00
CLR FSN3;сброс префикса защиты из S0_00
CLR BLWD;сброс префикса второго поля векторов клавиатуры
MOV R4,#<STR4;строка "- - - - " на индикацию
MOV R5,#>STR4
ACALL SINDI;до отработки первого цикла терморегулятора
ACALL LLCDD10;
AJMP PATCH;
;-----
SNWRI: MOV CNER,#3
CLR TR0
CLR TR1
CLR EA
TEST2: ACALL DTTC
ACALL T110
MOV A,#CCH
ACALL TXB
MOV A,#4EH
ACALL TXB
MOV A,HTPUD
ACALL TXB
MOV A,LTPUD
ACALL TXB
MOV A,CONF;#1FH только для DS18B20 - определяет
;режим формата данных в n бит по содержимому регистра CONF
ACALL TXB
ACALL DTTC
ACALL T110
MOV A,#CCH
ACALL TXB
MOV A,#BEH
ACALL TXB
RXDT2: MOV R0,#30H
MOV R5,#9
RXDT3: ACALL RXB
MOV @R0,A
INC R0
DJNZ R5,RXDT3
ACALL DTTC
ACALL T110

```



```

        ACALL CRCK
        JNB ERD,GOE0
        AJMP EERD;ошибка ответа датчика
GOE0:   JNB ERL,GOE1
        DJNZ CNER,TEST2
        AJMP EEE5;ошибка доверия CRC (зашумленная линия связи)
GOE1:   MOV A,HTPU
        CJNE A,HTPUD,ERWH;ошибка записи HTPUD
        MOV A,LTPU
        CJNE A,LTPUD,ERWL;ошибка записи LTPUD
        MOV A,CONF
        CJNE A,CONF,ERCFG;ошибка записи CONF
WEPROM: ACALL DTTC
        ACALL T110
        MOV A,#CCH
        ACALL TXB
        MOV A,#48H
        ACALL TXB
        ACALL DTTC
        ACALL T110
        SETB EA
        SETB TR0
        SETB TR1
        JNB ERD,NORMW
EERD:   SETB EROR
        AJMP PATCH
EEE5:   SETB EROR
        AJMP PATCH
ERCFG:  SETB EWFP
        SETB EROR
        AJMP PATCH
ERWL:   SETB EWFP
        SETB EROR
        AJMP PATCH
ERWH:   SETB EWFP
        SETB EROR
        AJMP PATCH
NORMW:  RET

```

```

;-----
REKS:   MOV CNER,#3
GUU2:   ACALL RDATU;читать текущую информацию из DS18B20
        JNB ERD,GUU1
        SJMP GUU3
GUU1:   JNB ECRC,GUU0
        DJNZ CNER,GUU2
GUU3:   SETB EROR
        AJMP PATCH
GUU0:   RET

```

```

;-----
RELY01: CLR CDT
        DJNZ CMSN,GOSK
        AJMP S3_10;прямой выход на п\п кнопки "С" второго поля
        ;клавиатуры (только в данном частном варианте!)
GOSK:   AJMP PATCH

```

```

;-----
; В полном варианте при "вложенных" модификациях кнопок вместо флага FSN1
; старшая тетрада
; указателя SNNK подставляется из младшей тетрады регистра TABSN
; (при BLWD=1) тем самым дается возможность получения до 256 указателей процедур
; обработки кнопок в пределах текущего содержимого DPTR ( по единой таблице
; VEKTS )
;-----

```

```

END

```